UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/775,732	02/09/2004	Bhasker Allam	010327-007210US	1552

20350 7590 10/22/2007

TOWNSEND AND TOWNSEND AND CREW, LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER	
ZHANG, SHIRLEY X	
ART UNIT	**PAPER NUMBER**
4121	
MAIL DATE	**DELIVERY MODE**
10/22/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/775,732	Applicant(s) ALLAM ET AL.
	Examiner Shirley X. Zhang	Art Unit 4121

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 February 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *02/09/2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

This non-final office action is responsive to the U.S. application No. 10/775,732 filed on 02/09/2004. It is acknowledged that the applicant has requested non-publication of the application file. The applicant has not filed an information disclosure statement.

Priority Claims

Acknowledgement is made of a claim for priority under 35 U.S.C. 119(e) to the U.S. provisional application No. 60/455,706 filed on 03/17/2003.

Specification

1. The abstract of the disclosure is objected to because the length of the abstract exceeds 150 words. Correction is required. See MPEP § 608.01(b).

2. The specification of the disclosure is objected to because of the following informalities:

There appears to be a mismatch between the reference label 32 and the term "virtual routers" recited in paragraph [0030] on page 7.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

3. **Claims 1 and 2** are rejected under 35 U.S.C. 102(a) as being anticipated by IP Infusion ("Virtual Routing for Provider Edge Applications", a white paper by **IP Infusion**, Inc.).

Regarding claim 1, IP Infusion teaches a routing device (Fig. 4 discloses a virtual router system) comprising:

a plurality of virtual routers (Fig. 4 discloses a plurality of virtual routers VR1, VR 2 and VRn that are situated inside the virtual router system);

a router manager configured to manage the plurality of virtual routers (Fig. 4 discloses a Global Management Authority (GMA) that creates and manages virtual routers); and

an application, wherein the application is situated external to the plurality of virtual routers (The Global Management Authority (GMA) disclosed in Fig. 4 is a software package that performs not only router manager functions, but also other system administrative functions such as remote login, therefore it is interpreted to anticipate said application recited in the claim.),

wherein the application is able to selectively communicate with one or more of the plurality of virtual routers on a dynamic basis to have the one or more virtual routers perform a plurality of tasks (Page 3, column 1 discloses that GMA is created so that global administrators can login to the router system in runtime and then selectively login to a chosen VR to execute commands).

Regarding claim 2, IP Infusion teaches the routing device of claim 1 wherein software is used to implement the plurality of virtual routers and the router manager (Page 5, column 2, paragraph 2 discloses that a virtual router comprises a full implementation of a physical router at the software level).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later

invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

4. **Claims 3-16** are rejected under 35 U.S.C. 103(a) as being unpatentable over IP Infusion, in view of Huang et al. ("The ENTRAPID Protocol Development Environment", hereinafter "**Huang**").

Regarding claim 3, IP Infusion teaches the routing device of claim 1 wherein each virtual router further includes:

a routing protocol stack configured to handle a plurality of routing protocols (Fig. 3 discloses a virtual router supporting a plurality of routing protocols including RIP, OSPF and BGP);

a plurality of interface drivers configured to communicate with a plurality of corresponding physical interfaces (Page 6, column 2, paragraph 2 discloses that multiple interfaces can be assigned to a single VR, where it is well known in the art that every physical interface inherently corresponds to an interface driver);

an Internet Protocol (IP) stack configured to interact with the routing protocol stack and perform a forwarding function via the plurality of interface drivers (Fig. 3 discloses that the virtual router contains a TCP/IP stack that interacts with routing protocols such as RIP, OSPF and BGP on one side, and connects to a forwarding plane to perform forwarding function via the physical interfaces on the other side, as is described in page 4, column 1, paragraph 2; The forwarding plane inherently forwards packets to the physical interfaces via interface drivers.), the IP stack having a forwarding information table, information from which is used to

perform the forwarding function (page 6, column 1, paragraph 3 discloses that the TCP/IP stack includes many features of a VR, such as software forwarding and management of the FIBs).

IP Infusion does not teach but Huang teaches a socket layer having a corresponding socket application programming interface, the socket layer configured to facilitate interactions between the IP stack and the routing protocol stack and the application, wherein the socket application programming interface is used to facilitate communications with the socket layer (Huang, page 4, column 1, paragraph 1 discloses that applications built using the BSD socket API can be ported immediately to a virtualized network kernel (VNK) which is a user-space IP protocol stack. Huang's disclosure implies that VNK supports a BSD socket API.)

It would have been obvious for one of ordinary skill in the art to apply Huang's teaching of socket API to IP Infusion's virtual router system such that a socket layer is included to facilitate interactions between the IP stack, the routing protocol stack and the application, as is recited in the claim. One would have been motivated to combine as such for the reason that sockets have been widely used in network programming as an inter-process communication mechanism since its introduction in the 1980s.

Regarding claim 4, the combination of IP Infusion and Huang teaches the routing device of claim 3.

IP Infusion does not teach that the routing device further comprises an operating system kernel wherein the IP stack of each of the plurality of virtual routers resides external to the operating system kernel.

However, Huang discloses a virtualized networking system that comprises an operating system kernel and a plurality of virtualized networking kernels (VNK) and processes residing

external to the OS kernel (see page 3, column 2). A VNK is the result of extracting the networking portion of the FreeBSD protocol from the kernel and moving it into the user space, i.e. a VNK is an instance of the user-space IP stack (see page 5, column 1, paragraph 2). Huang further discloses that one or more virtualized processes can run on top of a VNK to implement networking protocols such routing protocols above the IP layer, as the purpose of ENTRAPID is to provide a protocol development environment (see page 4, column 1, paragraph 1 and page 8, column 1, section VII). Thus, a VNK and its corresponding virtualized processes together form a virtual router.

Therefore, it would have been obvious for one of ordinary skill in the art to modify IP Infusion's virtual router system with Huang's teaching such that the routing device further comprises an operating system kernel wherein the IP stack of each of the plurality of virtual routers resides external to the operating system kernel. One would have been motivated to combine as such for the desirable advantages that (1) the resulted system allows each copy of the IP stack to work independently; (2) existing user-space applications can be ported immediately to the virtual router (Huang, page 4, column 1), and (3) developers can monitor and modify any aspect of the entire protocol stack without having to make any changes to the kernel (Huang, page 3, column 2, last paragraph).

Regarding claim 5, the combination of IP Infusion and Huang teaches the routing device of claim 4.

IP Infusion does not teach but Huang further teaches that the operating system kernel includes an associated socket layer, the socket layer having a corresponding socket application programming interface, and the application is able to communicate with the operating system

kernel via the associated socket layer using the corresponding socket application programming interface to have the operating system kernel perform one or more of the plurality of tasks (Huang, page 5, column 2, paragraph 1 discloses that ENTRAPID's virtualization approach has been tested on FreeBSD and can be further applied to Windows NT and Solaris. It is known in the art of networking that in FreeBSD, Solaris and Windows, the TCP/IP stack is implemented in the kernel, and is accessible to user space applications via a socket API, i.e., applications external to the operating system kernel communicate with the kernel via a socket layer and socket API).

Therefore, it would have been obvious for one of ordinary skill in the art to modify IP Infusion's virtual router system with Huang's teaching so that the operating system kernel of the virtual router system includes an associated socket layer. The fact that socket API has been used as a standard way of communication between networking applications and kernel since the 1980s would have motivated one of ordinary skill in the art to make such combination at the time the invention was made.

Regarding claim 6, the combination of IP Infusion and Huang teaches the routing device as recited in claim 1. IP Infusion further teaches an UNIX system incorporating the routing device (page 5, column 1, paragraph 1 discloses that the virtual router system is implemented on Linux, which is a UNIX operating system).

Regarding claim 7, IP Infusion teaches a routing device comprising:

an operating system kernel (page 5, column 1, paragraph 1 discloses that the virtual router is implemented on Linux, VxWorks and OSE, all of which contain an operating system kernel);

a virtual router (Fig. 3 discloses the virtual router 1 in a virtual router system),

a router manager configured to manage the virtual router (Fig. 4 discloses a Global Management Authority (GMA) that creates and manages virtual routers);

an application residing external to the virtual router (The Global Management Authority (GMA) disclosed in Fig. 4 is a software package that performs not only router manager functions, but also other system administrative functions such as remote login, therefore it is interpreted to anticipate said application recited in the claim); and

a plurality of physical interfaces (Fig. 5 and page 3, column 1, paragraph 1 disclose that the routing device includes multiple physical interfaces);

wherein the application is able to selectively interact with the virtual router and the operating system kernel on a dynamic basis in order to have the virtual router and the operating system kernel perform a plurality of tasks for the application (Page 3, column 1 discloses that GMA is created so that global administrators can login to the router system in runtime and then selectively login to a chosen VR to execute commands).

IP Infusion does not teach that the virtual router resides external to the operating system kernel.

However, Huang discloses a virtualized networking system that comprises an operating system kernel and a plurality of virtualized networking kernels (VNK) and processes residing external to the OS kernel (see page 3, column 2). A VNK is the result of extracting the networking portion of the FreeBSD protocol from the kernel and moving it into the user space, i.e. a VNK is an instance of the user-space IP stack (see page 5, column 1, paragraph 2). Huang further discloses that one or more virtualized processes can run on top of a VNK to implement networking protocols such routing protocols above the IP layer, as the purpose of ENTRAPID is

to provide a protocol development environment (see page 4, column 1, paragraph 1 and page 8, column 1, section VII). Thus, a VNK and its corresponding virtualized processes together form a virtual router.

It would have been obvious for one of ordinary skill in the art to modify IP Infusion's virtual router system with Huang's teaching so that the routing device further comprises an operating system kernel wherein the virtual routers resides external to the operating system kernel. One would have been motivated to combine as such for the desirable advantages that (1) the resulted system allows each copy of the IP stack to work independently; (2) existing user-space applications can be ported immediately to the virtual router (Huang, page 4, column 1), and (3) developers can monitor and modify any aspect of the entire protocol stack without having to make any changes to the kernel (Huang, page 3, column 2, last paragraph).

Regarding claim 8, the combination of IP Infusion and Huang teaches the routing device of claim 7. IP Infusion further teaches that software is used to implement the virtual router and the router manager (Page 5, column 2, paragraph 2 discloses that a virtual router comprises a full implementation of a physical router at the software level).

Regarding claim 9, the combination of IP Infusion and Huang teaches the routing device of claim 7. IP Infusion further teaches that the virtual router includes:

a routing protocol stack configured to handle a plurality of routing protocols (Fig. 3 discloses a virtual router supporting a plurality of routing protocols including RIP, OSPF and BGP);

a plurality of interface drivers configured to communicate with corresponding physical interfaces (Page 6, column 2, paragraph 2 discloses that multiple interfaces can be assigned to a

single VR, where it is well known in the art that every physical interface inherently corresponds to an interface driver);

an Internet Protocol (IP) stack configured to interact with the routing protocol stack and perform a forwarding function via the plurality of interface drivers (Fig. 3 discloses that the virtual router contains a TCP/IP stack that interacts with routing protocols such as RIP, OSPF and BGP on one side, and connects to a forwarding plane to perform forwarding function via the physical interfaces on the other side, as is described in page 4, column 1, paragraph 2; The forwarding plane inherently forwards packets to the physical interfaces via interface drivers.), the IP stack having a forwarding information table, information from which is used to perform the forwarding function (page 6, column 1, paragraph 3 discloses that the TCP/IP stack includes many features of a VR, such as software forwarding and management of the FIBs); and

IP Infusion does not teach but Huang teaches a socket layer having a corresponding socket application programming interface, the socket layer configured to facilitate interactions between the IP stack and the routing protocol stack and the application, wherein the socket application programming interface is used to facilitate communications with the socket layer (Huang, page 4, column 1, paragraph 1 discloses that applications built using the BSD socket API can be ported immediately to a virtualized network kernel (VNK) which is a user-space IP protocol stack. Huang's disclosure implies that VNK supports a BSD socket API.)

It would have been obvious for one of ordinary skill in the art to apply Huang's teaching of socket API to IP Infusion's virtual router system such that a socket layer is included to facilitate interactions between the IP stack, the routing protocol stack and the application, as is recited in the claim. One would have been motivated to combine as such for the reason that

sockets have been widely used in network programming as an inter-process communication mechanism since its introduction in the 1980s.

Regarding claim 10, the combination of IP Infusion and Huang teaches the routing device of claim 9.

IP Infusion does not teach that the IP stack of the virtual router resides external to the operating system kernel.

However, as mentioned above in claim 7, Huang discloses a virtualized networking system that comprises an operating system kernel and a plurality of virtual routers residing external to the OS kernel (see page 3, column 2). Each virtual router comprises one VNK and a plurality of corresponding virtualized processes, where the VNK is a user-space IP stack.

It would have been obvious for one of ordinary skill in the art to modify IP Infusion's virtual router system with Huang's teaching such that the IP stack of the virtual router resides external to the operating system kernel. One would have been motivated to combine as such for the desirable advantages that (1) the resulted system allows each copy of the IP stack to work independently; 2) existing user-space applications can be ported immediately to the virtual router (Huang, page 4, column 1), and (3) developers can monitor and modify any aspect of the entire protocol stack without having to make any changes to the kernel (Huang, page 3, column 2, last paragraph).

Regarding claim 11, the combination of IP Infusion and Huang teaches the routing device of claim 7.

IP Infusion does not teach but Huang further teaches that the operating system kernel includes an associated socket layer, the socket layer having a corresponding socket application

programming interface, and the application is able to communicate with the operating system kernel via the associated socket layer using the corresponding socket application programming interface to have the operating system kernel perform one or more of the plurality of tasks (Huang, page 5, column 2, paragraph 1 discloses that ENTRAPID's virtualization approach has been tested on FreeBSD and can be further applied to Windows NT and Solaris. It is known in the art of networking that in FreeBSD, Solaris and Windows, the TCP/IP stack is implemented in the kernel, and is accessible to user space applications via a socket API, i.e., applications external to the operating system kernel communicate with the kernel via a socket layer and a socket API).

Therefore, it would have been obvious for one of ordinary skill in the art to modify IP Infusion's virtual router system with Huang's teaching so that the operating system kernel of the virtual router system includes an associated socket layer. The fact that socket API has been used as a standard way of communication between networking applications and kernel since the 1980s would have motivated one of ordinary skill in the art to make such combination at the time the invention was made.

Regarding claim 12, the combination of IP Infusion and Huang teaches the routing device as recited in claim 7. IP Infusion further teaches an UNIX system incorporating the routing device (page 5, column 1, paragraph 1 discloses that the virtual router system is implemented on Linux, which is a UNIX operating system).

Regarding claim 13, IP Infusion teaches a routing device comprising a plurality of virtual routers (Fig. 4 discloses a plurality of virtual routers VR1, VR 2 and VRn that are situated inside the virtual router system), and

an application residing external to the plurality of virtual routers (The Global Management Authority (GMA) disclosed in Fig. 4 is also an application situated external to the plurality of virtual routers; As the instant invention does not disclose the functions of said application, IP Infusion's GMA is interpreted to anticipate said application recited in the claim);

wherein the application is able to selectively interact with one of the plurality of virtual routers (Page 3, column 1 discloses that GMA is created so that global administrators can login to the router system in runtime and then selectively login to a chosen VR to execute commands).

IP Infusion does not disclose that each virtual router has an associated socket layer and an Internet Protocol (IP) stack, wherein the associated socket layer has a corresponding socket application programming interface configured to facilitate communications with the associated socket layer, and the associated socket layer is configured to facilitate interactions between the IP stack and the application

However, Huang discloses in page 4, column 1, paragraph 1 that applications built using the BSD socket API can be ported immediately to a virtualized network kernel (VNK) which is a user-space IP stack. Huang's disclosure implies that the VNK supports a BSD socket API, through which user space applications can communicate with the socket layer and the IP stack.

It would have been obvious for one of ordinary skill in the art to apply Huang's teaching of socket API to IP Infusion's virtual router system such that a socket layer and API is included to facilitate interactions between the IP stack and the application, as is recited in the claim. One would have been motivated to combine as such because sockets have been widely used in

network programming as an inter-process communication mechanism since its introduction in the 1980s.

Regarding claim 14, the combination of IP Infusion and Huang teaches the routing device of claim 13. IP Infusion further teaches that the routing device comprises:

an operating system kernel (page 5, column 1, paragraph 1 discloses that the virtual router is implemented on Linux, VxWorks and OSE, all of which contain an operating system kernel);

wherein the application is able to selectively interact with one or more of the plurality of virtual routers on a dynamic basis in order to have one or more of the plurality of virtual routers perform a plurality of tasks for the application (page 3, column 1 discloses that GMA is created so that global administrators can login to the router system in runtime and then selectively login to a chosen VR to execute commands).

IP Infusion does not teach but Huang teaches that the application is able to interact with the operating system kernel to have the operating system kernel perform a plurality of tasks for the application (page 5, column 2, last paragraph discloses that ENTRAPID employs proxy virtualized processes to facilitate the communication between processes external to ENTRAPID and the operating system kernel. The processes external to ENTRAPID are equivalent to the said applications recited in the claim.

It would have been obvious to one of ordinary skill in the art to combine the teachings of IP Infusion and Huang to allow the application external to the virtual router to interact with the operating system kernel. The need to access non-virtualized shared resources that are controlled

exclusively by the kernel such as the file system would have motivated one of ordinary skill to make such combination.

Regarding claim 15, the combination of IP Infusion and Huang teaches the routing device of claim 13.

IP Infusion does not teach that the plurality of virtual routers reside external to the operating system kernel.

However, Huang discloses in page 3, column 2 a virtualized networking system that comprises an operating system kernel, and a plurality of virtualized networking kernels (VNK) and virtualized processes that reside external to the OS kernel. As is further disclosed in page 5, column 1, paragraph 2, a VNK is the result of extracting the networking portion of the FreeBSD protocol from the kernel and moving it into the user space, therefore, a VNK is an instance of the user-space IP stack.

Huang further discloses that one or more virtualized processes can run on top of a VNK to implement networking protocols such routing protocols above the IP layer (see page 4, column 1, paragraph 1 and page 8, column 1, section VII). Thus, a VNK and its corresponding virtualized processes together form a virtual router.

It would have been obvious for one of ordinary skill in the art to modify IP Infusion's virtual router system with Huang's teaching such that the IP stack of the virtual router resides external to the operating system kernel. One would have been motivated to combine as such for the desirable advantages that (1) the resulted system allows each copy of the IP stack to work independently; (2) existing user-space applications can be ported immediately to the virtual router (Huang, page 4, column 1), and (3) developers can monitor and modify any aspect of the

entire protocol stack without having to make any changes to the kernel (Huang, page 3, column 2, last paragraph).

Regarding claim 16, the combination of IP Infusion and Huang teaches the routing device as recited in claim 13. IP Infusion further teaches an UNIX system incorporating the routing device (page 5, column 1, paragraph 1 discloses that the virtual router system is implemented on Linux, which is a UNIX operating system).

Conclusion

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent No. 7,242,665, Paul F. Langille et al., "Network Device Virtual Interface";

U.S. Patent No. 7,039,720, Stephen M. Alfieri et al., "Dense Virtual Router Packet Switching";

U.S. Patent No. 6,907,039, Naiming Shen, "Method and Apparatus for Routing And Forwarding Between Virtual Routers Within a Single Network Element";

U.S. Patent Publication 2003/0051048, Thomas Lee Watson, "System and Method For Router Virtual Networking";

Jain, P.G.; Hutchinson, N.C.; Chanson, S.T., "A Framework for the Non-Monolithic Implementation of Protocols in the x-Kernel", USENIX, August 1994, High-Speed Networking Symposium, pp. 13-30

C. A. Thekkath, T. D. Nguyen, E. Moy, and E. D. Lazowska, "Implementing Network Protocols at User Level", IEEE/ACM Transactions on Networking, 1(5): 554-565, Nov 1993.

Jeffrey C. Mogul, Richard F. Rashid, and Michael J. Accetta. "The packet filter: An efficient mechanism for user-level network code". In Proceedings of the Eleventh ACM Symposium on Operating Systems Principles, pages 39-51. ACM Press, November 1987

Chris Maeda and Brian N. Bershad. "Protocol service decomposition for high-performance networking", In Proceedings 14th SOSP, pages 244-255, Asheville, NC, USA, Dec 1993.

M. Zec. "Implementing a Clonable Network Stack in the FreeBSD Kernel", In Proceedings of the USENIX 2003

D. Ely, S. Savage, and D. Wetherall, "Alpine: A User-Level infrastructure for network protocol development," in Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems, 2001, pp. 171-184.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shirley X. Zhang whose telephone number is (571) 270-5012. The examiner can normally be reached on Monday through Friday 7:30am - 5:00pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Taghi Arani can be reached on (571) 272-3787. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Taghi T. Arani

TAGHI ARANI
PRIMARY EXAMINER

10/17/07